Out Of Process Execution for Clang-Repl

Gsoc 2024 by Sahil Patidar
Mentor : Vassil Vassilev, Matheus Izvekov

Midterm Progress Update

Implementation Out-of-Process Execution for Clang-Repl

Implemented out-of-process execution: Added two new flags, `oop-executor` and `oop-executor-connect`, to Clang-Repl. These allow the REPL to launch an external executor (`llvm-jitlink-executor`) to handle code execution in a separate process.
Result: With these changes, Clang-Repl can now run code in a separate process on Mach-O, reducing resource usage and preventing crashes in the main program when user code fails.

ORC Enhancements - dlupdate for Mach-O

Implemented the `dlupdate` function: This new function in the ORC runtime enables incremental execution of initializers, which are essential for handling new code added during REPL sessions.

Difference from `dlopen`: Unlike `dlopen`, which manages code mapping and library references, `dlupdate` focuses only on running new initializers, making the process more efficient.

Next Step: A similar `dlupdate` implementation is needed for ELF to ensure it supports incremental initializers like Mach-O.

Issues Encountered

ELF Limitations

The current ELF design does not retain initializers after dlopen runs. Once initializers are processed, they are erased, which prevents re-execution. This makes ELF unsuitable for Clang-Repl incremental execution.

Current State

Mach-O: The out-of-process execution is functioning correctly for Mach-O, with incremental initializers working as intended.
ELF: Initializers are not preserved in ELF after they are loaded, which affects the ability to re-run initializers.

Next Steps

Redesign ELF Initializer Handling

1. Redesign ELF Initializer Handling : Adapt ELF to use a push-record model similar to Mach-O. Introduce push-initializers and record-initializers operations, where the `JITDylibState` stores initializers in ORC runtime state.
Use the Mach-O `RecordSectionsTracker` class template to store and manage recorded initializers, allowing us to run only the new initializers using `processNewSections`.

2. Move to Alloc-Actions for ELF : Transition ELF from multiple RPC calls to a more efficient alloc-actions model. This will reduce the number of RPC calls needed for memory allocations and include cleanup actions within the same call, simplifying resource management and allowing the JIT code to be cleaned up automatically.

Conclusion

Significant progress has been made in enabling out-of-process execution for Clang-Repl, especially on Mach-O. The next major steps involve addressing ELF limitations, including implementing `dlupdate` for ELF, to ensure it supports incremental execution properly.

Thank You!